```
EEEEEEEEEEEEEEE    RRRRRRRRRRRR        FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE    RRRRRRRRRRRRR       FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE    RRRRRRRRRRRRR       FFFFFFFFFFFFFF
EEE                RRR        RRR      FFF
EEE                RRR        RRR      FFF
EEE                RRR        RRR      FFF
EEE                RRR        RRR      FFF
EEE                RRR        RRR      FFF
EEE                RRR        RRR      FFF
EEEEEEEEEEEE       RRRRRRRRRRRR        FFFFFFFFFFF
EEEEEEEEEEEE       RRRRRRRRRRRR        FFFFFFFFFFF
EEEEEEEEEEEE       RRRRRRRRRRRR        FFFFFFFFFFF
EEE                RRR   RRR           FFF
EEE                RRR   RRR           FFF
EEE                RRR   RRR           FFF
EEE                RRR       RRR       FFF
EEE                RRR       RRR       FFF
EEE                RRR       RRR       FFF
EEEEEEEEEEEEEEE    RRR          RRR    FFF
EEEEEEEEEEEEEEE    RRR          RRR    FFF
EEEEEEEEEEEEEEE    RRR          RRR    FFF
```

```
VV       VV    AAAAAA     XX    XX  77777777      888888        000000     RRRRRRRR    EEEEEEEEEE     GGGGGGGG
VV       VV    AAAAAA     XX    XX  77777777      888888        000000     RRRRRRRR    EEEEEEEEEE     GGGGGGGG
VV       VV    AA    AA   XX    XX        77    88      88    00      00    RR      RR  EE                  GG
VV       VV    AA    AA   XX    XX        77    88      88    00      00    RR      RR  EE                  GG
VV       VV    AA    AA    XX  XX         77    88      88    00    0000    RR      RR  EE                  GG
VV       VV    AA    AA    XX  XX         77    88      88    00    0000    RR      RR  EE                  GG
VV       VV    AA    AA      XX           77      888888      00  00  00    RRRRRRRR    EEEEEEE             GG
VV       VV    AA    AA      XX           77      888888      00  00  00    RRRRRRRR    EEEEEEE             GG
VV       VV    AAAAAAAAAA   XX  XX        77    88      88    0000    00    RR    RR     EE          GG   GGGGG
VV       VV    AAAAAAAAAA   XX  XX        77    88      88    0000    00    RR    RR     EE          GG   GGGGG
  VV   VV      AA    AA    XX    XX       77    88      88    00      00    RR      RR   EE          GG      GG
  VV   VV      AA    AA    XX    XX       77    88      88    00      00    RR      RR   EE          GG      GG
    VV         AA    AA    XX    XX  77          888888        000000      RR      RR   EEEEEEEEEE     GGGGG
    VV         AA    AA    XX    XX  77          888888        000000      RR      RR   EEEEEEEEEE     GGGGG


LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

```
0001    C
0002    C Version:      'V04-000'
0003    C
0004    C********************************************************************
0005    C*                                                                  *
0006    C*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                        *
0007    C*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.         *
0008    C*   ALL RIGHTS RESERVED.                                           *
0009    C*                                                                  *
0010    C*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0011    C*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0012    C*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0013    C*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0014    C*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0015    C*   TRANSFERRED.                                                    *
0016    C*                                                                  *
0017    C*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0018    C*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0019    C*   CORPORATION.                                                    *
0020    C*                                                                  *
0021    C*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0022    C*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.        *
0023    C*                                                                  *
0024    C*                                                                  *
0025    C********************************************************************
0026    C
0027
0028            SUBROUTINE VAX780REG (LUN,REGISTER)
0029
0030    C
0031    C       AUTHOR  BRIAN PORTER             CREATION DATE   9-AUG-1979
0032    C
0033    C       Functional desciption:
0034    C
0035    C       This module displays all the Kernel registers that appear in various
0036    C       error log entries for the 11/780.
0037    C
0038    C       Modified by:
0039    C
0040    C       V03-006 SAR0201         Sharon A. Reynolds,     27-Feb-1984
0041    C               Added an SYE update that fixes a bug in the
0042    C               SBI confirmation codes in SBIERR.
0043    C
0044    C       V03-005 SAR0162         Sharon A. Reynolds,     13-Oct-1983
0045    C               Added an SYE update that adds 11/7XX support.
0046    C
0047    C       V03-004 SAR0106         Sharon A. Reynolds,     20-Jun-1983
0048    C               Changed the carriage control in the 'format' statements
0049    C               for use with ERF.
0050    C
0051    C       v03-003 BP0012          Brian Porter,           27-FEB-1983
0052    C               Enhanced sbi silo output.
0053    C
0054    C       v03-002 BP0011          Brian Porter,           19-AUG-1982
0055    C               Corrected ms780e memory size.
0056    C
0057    C       v03-001 BP0010          Brian Porter,           19-FEB-1982
```

```
0058        C          Added ci780 support.
0059        C**
0060
0061
0062        BYTE          LUN
0063
0064        INTEGER*4     REGISTER
0065
0066        INTEGER*4     FIELD
0067
0068        INTEGER*4     COMPRESSC
0069
0070        integer*4     compress4
0071
0072        integer*4     lib$extzv
0073
0074        INTEGER*4     FIELD1
0075
0076        PARAMETER     FOUR_K = 1
0077
0078        PARAMETER     SIXTEEN_K = 2
0079
0080        PARAMETER     MA780_0 = 64
0081
0082        PARAMETER     MA780_3 = 67
0083
0084        PARAMETER     UBA_0 = 40
0085
0086        PARAMETER     UBA_3 = 43
0087
0088        PARAMETER     MS780_4K = 8
0089
0090        PARAMETER     MS780_4KI = 9
0091
0092        PARAMETER     MS780_16K = 16
0093
0094        PARAMETER     MS780_16KI = 17
0095
0096        PARAMETER     DR780 = 48
0097
0098        PARAMETER     MBA = 32
0099
0100        PARAMETER     COMMAND_ADDRESS = 3
0101
0102        PARAMETER     READ_DATA = 0
0103
0104        CHARACTER*21  V1UBA_REGA(16:18)
0105
0106        CHARACTER*17  V1MS780C_REGA(0:0)
0107
0108        character*17  v1ms780e_rega(8:8)
0109
0110        equivalence   (v1ms780c_rega,v1ms780e_rega)
0111
0112        character*30  v2ms780e_rega(15:20)
0113
0114        CHARACTER*26  MS780C_RAM_TYPE(0:3)
```

```
0115
0116          character*26    ms780e_ram_type(0:3)
0117
0118          character*31    ms780e_interleave_mode(0:4)
0119
0120          CHARACTER*24    V2DRCR(11:11)
0121
0122          CHARACTER*21    V3DRCR(15:20)
0123
0124          CHARACTER*15    V4DRCR(24:24)
0125
0126          CHARACTER*25    V1DRCR(1:3)
0127
0128          CHARACTER*12    SBI_CONFIRM(1:3)
0129
0130          CHARACTER*22    V1SBI_ERROR(1:3)
0131
0132          CHARACTER*23    V2SBI_ERROR(7:8)
0133
0134          CHARACTER*25    V3SBI_ERROR(13:15)
0135
0136          CHARACTER*29    V1TIMEOUT_ADDR(29:29)
0137
0138          CHARACTER*15    ACCS_TYPE(0:2)
0139
0140          CHARACTER*20    V1ACCS(15:15)
0141
0142          CHARACTER*23    V2ACCS(27:29)
0143
0144          CHARACTER*6     V3ACCS(31:31)
0145
0146          CHARACTER*23    V1SBI_FAULT(16:19)
0147
0148          CHARACTER*31    V2SBI_FAULT(26:31)
0149
0150          CHARACTER*31    V1SBI_REGA(21:23)
0151
0152          CHARACTER*31    V2SBI_REGA(26:31)
0153
0154          CHARACTER*27    V1SBI_COMPARATR(29:31)
0155
0156          CHARACTER*22    TIMEOUT_STATUS(0:3)
0157
0158          CHARACTER*22    IB_STATUS(0:3)
0159
0160          CHARACTER*22    CP_STATUS(0:3)
0161
0162          CHARACTER*11    REF_MODE(0:3)
0163
0164          CHARACTER*21    SBI_RESPONSE(0:2)
0165
0166          CHARACTER*17    V1SBI_SILO(30:31)
0167
0168          CHARACTER*18    SBI_TAG(0:7)
0169
0170          CHARACTER*25    COND_LOCK(1:3)
0171
```

```
0172              CHARACTER*23    SBI_FUNCTION(0:12)
0173
0174              character*19    v1ci780_rega(8:10)
0175
0176              character*25    v2ci780_rega(16:20)
0177
0178              EQUIVALENCE     (V2SBI_REGA,V2SBI_FAULT)
0179
0180              EQUIVALENCE     (TIMEOUT_STATUS,IB_STATUS)
0181
0182              EQUIVALENCE     (TIMEOUT_STATUS,CP_STATUS)
0183
0184
0185
0186       DATA     V1UBA_REGA(16)  /'UNIBUS INIT COMPLETE*'/
0187
0188       DATA     V1UBA_REGA(17)  /'UNIBUS POWER DOWN*'/
0189
0190       DATA     V1UBA_REGA(18)  /'UNIBUS INIT ASSERTED*'/
0191
0192
0193
0194       DATA     MS780C_RAM_TYPE(0)      /'NO ARRAY BOARDS PRESENT*'/
0195
0196       DATA     MS780C_RAM_TYPE(1)      /'4K RAM ARRAY BOARDS*'/
0197
0198       DATA     MS780C_RAM_TYPE(2)      /'16K RAM ARRAY BOARDS*'/
0199
0200       DATA     MS780C_RAM_TYPE(3)      /'MULTIPLE ARRAY TYPE ERROR*'/
0201
0202
0203
0204       data     ms780e_ram_type(0)      /'MULTIPLE ARRAY TYPE ERROR*'/
0205
0206       data     ms780e_ram_type(1)      /'64K RAM ARRAY BOARDS*'/
0207
0208       data     ms780e_ram_type(2)      /'256K RAM ARRAY BOARDS*'/
0209
0210       data     ms780e_ram_type(3)      /'NO ARRAY BOARDS PRESENT*'/
0211
0212
0213
0214
0215       DATA     V1MS780C_REGA(0)/'INTERLEAVED MODE*'/
0216
0217
0218
0219
0220       data     v2ms780e_rega(15)       /'LOWER MISCONFIGURATION*'/
0221
0222       data     v2ms780e_rega(16)       /'UPPER MISCONFIGURATION*'/
0223
0224       data     v2ms780e_rega(17)       /'INTERLEAVE MISCONFIGURATION*'/
0225
0226       data     v2ms780e_rega(18)       /'LOWER CONTROLLER PARITY ERROR*'/
0227
0228       data     v2ms780e_rega(19)       /'UPPER CONTROLLER PARITY ERROR*'/
```

```
0229
0230          data    v2ms780e_rega(20)        /'ERROR SUMMARY*'/
0231
0232
0233
0234
0235          data    ms780e_interleave_mode(0)
0236        1 /'NON-INTERLEAVED (LOWER)*'/
0237
0238          data    ms780e_interleave_mode(1)
0239        1 /'EXTERNALLY-INTERLEAVE-(LOWER)*'/
0240
0241          data    ms780e_interleave_mode(2)
0242        1 /'NON-INTERLEAVED (UPPER)*'/
0243
0244          data    ms780e_interleave_mode(3)
0245        1 /'EXTERNALLY-INTERLEAVED (UPPER)*'/
0246
0247          data    ms780e_interleave_mode(4)
0248        1 /'INTERNALLY-2-WAY INTERLEAVED*'/
0249
0250
0251
0252
0253
0254          DATA    V1DRCR(1)        /'INTERLOCK SEQUENCE FAULT*'/
0255
0256          DATA    V1DRCR(2)        /'READ DATA TIMEOUT FAULT*'/
0257
0258          DATA    V1DRCR(3)        /'ILLEGAL TIMEOUT STATUS*'/
0259
0260
0261
0262          DATA    V2DRCR(11)       /'DDI DATA STALL*'/
0263
0264
0265
0266          DATA    V3DRCR(15)       /'READ DATA SUBSTITUTE*'/
0267
0268          DATA    V3DRCR(16)       /'CORRECTED READ DATA*'/
0269
0270          DATA    V3DRCR(17)       /'MICRO-CODE HALTED*'/
0271
0272          DATA    V3DRCR(18)       /'ABORT*'/
0273
0274          DATA    V3DRCR(19)       /'PACKET INTERRUPT*'/
0275
0276          DATA    V3DRCR(20)       /'INTERRUPT ENABLE*'/
0277
0278
0279
0280          DATA    V4DRCR(24)       /'EXTERNAL ABORT*'/
0281
0282
0283
0284          DATA    SBI_CONFIRM(1)  /'ACKNOWLEDGE*'/
0285
```

```
0286          DATA    SBI_CONFIRM(2)    /'BUSY*'/
0287
0288          DATA    SBI_CONFIRM(3)    /'ERROR*'/
0289
0290
0291
0292          DATA    V1SBI_ERROR(1)    /'SBI NOT BUSY*'/
0293
0294          DATA    V1SBI_ERROR(2)    /'MULTIPLE CPU ERROR*'/
0295
0296          DATA    V1SBI_ERROR(3)    /'IB ERROR CONFIRMATION*'/
0297
0298
0299
0300
0301          DATA    V2SBI_ERROR(7)    /'IB RECEIVED RDS*'/
0302
0303          DATA    V2SBI_ERROR(8)    /'CPU ERROR CONFIRMATION*'/
0304
0305
0306
0307          DATA    V3SBI_ERROR(13)   /'RDS CONFIRMATION*'/
0308
0309          DATA    V3SBI_ERROR(14)   /'CRD CONFIRMATION*'/
0310
0311          DATA    V3SBI_ERROR(15)   /'RDS/CRD INTERRUPT ENABLE*'/
0312
0313
0314
0315
0316
0317
0318          DATA    V1TIMEOUT_ADDR    /'PROTECTION CHECKED REFERENCE*'/
0319
0320
0321
0322          DATA    ACCS_TYPE(0)      /'NOT PRESENT*'/
0323
0324          DATA    ACCS_TYPE(1)      /'FLOATING POINT*'/
0325
0326          DATA    ACCS_TYPE(2)      /'UNKNOWN*'/
0327
0328
0329
0330          DATA    V1ACCS(15)        /'ACCELERATOR ENABLED*'/
0331
0332
0333
0334          DATA    V2ACCS(27)        /'RESERVED OPERAND ERROR*'/
0335
0336          DATA    V2ACCS(28)        /'OVERFLOW ERROR*'/
0337
0338          DATA    V2ACCS(29)        /'UNDERFLOW ERROR*'/
0339
0340
0341
0342          DATA    V3ACCS(31)        /'ERROR*'/
```

```
0343
0344
0345
0346
0347        DATA    V1SBI_FAULT(16) /'FAULT SILO LOCK*'/
0348
0349        DATA    V1SBI_FAULT(17) /'SBI FAULT*'/
035C
0351        DATA    V1SBI_FAULT(18) /'FAULT INTERRUPT ENABLE*'/
0352
0353        DATA    V1SBI_FAULT(19) /'FAULT LATCH*'/
0354
0355
0356
0357
0358
0359        DATA    V1SBI_REGA(21)/'ADAPTER OVER-TEMPERATURE*'/
0360
0361        DATA    V1SBI_REGA(22)/'ADAPTER POWER-UP*'/
0362
0363        DATA    V1SBI_REGA(23)/'ADAPTER POWER-DOWN*'/
0364
0365
0366
0367
0368        DATA    V2SBI_REGA(26)/'TRANSMITTER DURING FAULT CYCLE*'/
0369
0370        DATA    V2SBI_REGA(27)/'MULTIPLE TRANSMITTER FAULT*'/
0371
0372        DATA    V2SBI_REGA(28)/'INTERLOCK SEQUENCE FAULT*'/
0373
0374        DATA    V2SBI_REGA(29)/'UNEXPECTED READ DATA FAULT*'/
0375
0376        DATA    V2SBI_REGA(30)/'WRITE SEQUENCE FAULT*'/
0377
0378        DATA    V2SBI_REGA(31)/'PARITY FAULT*'/
0379
0380
0381
0382
0383        DATA    V1SBI_COMPARATR(29)/'LOCK UNCONDITIONAL*'/
0384
0385        DATA    V1SBI_COMPARATR(30)/'SILO LOCK INTERRUPT ENABLE*'/
0386
0387        DATA    V1SBI_COMPARATR(31)/'COMPARATOR SILO LOCK*'/
0388
0389
0390
0391        DATA    TIMEOUT_STATUS(0)/'NO RESPONSE*'/
0392
0393        DATA    TIMEOUT_STATUS(1)/'DEVICE BUSY*'/
0394
0395        DATA    TIMEOUT_STATUS(2)/'WAITING FOR READ DATA*'/
0396
0397        DATA    TIMEOUT_STATUS(3)/'ILLEGAL*'/
0398
0399
```

```
0400
0401
0402          DATA    REF_MODE(0)      /'KERNEL*'/
0403
0404          DATA    REF_MODE(1)      /'EXECUTIVE*'/
0405
0406          DATA    REF_MODE(2)      /'SUPERVISOR*'/
0407
0408          DATA    REF_MODE(3)      /'USER*'/
0409
0410
0411
0412
0413          DATA    SBI_RESPONSE(0) /'ERROR FREE DATA*'/
0414
0415          DATA    SBI_RESPONSE(1) /'CORRECTED READ DATA*'/
0416
0417          DATA    SBI_RESPONSE(2) /'READ DATA SUBSTITUTE*'/
0418
0419
0420
0421
0422
0423          DATA    V1SBI_SILO(30) /'SBI INTERLOCKED*'/
0424
0425          DATA    V1SBI_SILO(31) /'FAULT CLEAR FLAG*'/
0426
0427
0428
0429
0430          DATA    SBI_TAG(0)       /'READ DATA*'/
0431
0432          DATA    SBI_TAG(1)       /'ILLEGAL TAG*'/
0433
0434          DATA    SBI_TAG(2)       /'ILLEGAL TAG*'/
0435
0436          DATA    SBI_TAG(3)       /'COMMAND ADDRESS*'/
0437
0438          DATA    SBI_TAG(4)       /'ILLEGAL TAG*'/
0439
0440          DATA    SBI_TAG(5)       /'WRITE DATA*'/
0441
0442          DATA    SBI_TAG(6)       /'INTERRUPT SUMMARY*'/
0443
0444          DATA    SBI_TAG(7)       /'ILLEGAL TAG*'/
0445
0446
0447
0448
0449          DATA    COND_LOCK(1)     /'ID ONLY*'/
0450
0451          DATA    COND_LOCK(2)     /'ID AND TAG*'/
0452
0453          DATA    COND_LOCK(3)     /'ID, TAG AND COMMAND/MASK*'/
0454
0455
0456
```

```
0457
0458          DATA    SBI_FUNCTION(0) /'ILLEGAL FUNCTION*'/
0459
0460          DATA    SBI_FUNCTION(1) /'READ MASKED*'/
0461
0462          DATA    SBI_FUNCTION(2) /'WRITE MASKED*'/
0463
0464          DATA    SBI_FUNCTION(3) /'ILLEGAL FUNCTION*'/
0465
0466          DATA    SBI_FUNCTION(4) /'INTERLOCK READ MASKED*'/
0467
0468          DATA    SBI_FUNCTION(5) /'ILLEGAL FUNCTION*'/
0469
0470          DATA    SBI_FUNCTION(6) /'ILLEGAL FUNCTION*'/
0471
0472          DATA    SBI_FUNCTION(7) /'INTERLOCK WRITE MASKED*'/
0473
0474          DATA    SBI_FUNCTION(8) /'EXTENDED READ*'/
0475
0476          DATA    SBI_FUNCTION(9) /'ILLEGAL FUNCTION*'/
0477
0478          DATA    SBI_FUNCTION(10)/'ILLEGAL FUNCTION*'/
0479
0480          DATA    SBI_FUNCTION(11)/'EXTENDED WRITE MASKED*'/
0481
0482          DATA    SBI_FUNCTION(12)/'ILLEGAL FUNCTION*'/
0483
0484
0485
0486          data    v1ci780_rega(8) /'POWER-FAIL DISABLE*'/
0487
0488          data    v1ci780_rega(9) /'TRANSMIT DEAD*'/
0489
0490          data    v1ci780_rega(10)/'TRANSMIT FAIL*'/
0491
0492
0493
0494
0495          data    v2ci780_rega(16)/'CORRECTED READ DATA*'/
0496
0497          data    v2ci780_rega(17)/'READ DATA SUBSTITUTE*'/
0498
0499          data    v2ci780_rega(18)/'COMMAND TRANSMIT ERROR*'/
0500
0501          data    v2ci780_rega(19)/'READ DATA TIMEOUT*'/
0502
0503          data    v2ci780_rega(20)/'COMMAND TRANSMIT TIMEOUT*'/
0504
0505
0506
0507
0508          ENTRY ACCS_780 (LUN,REGISTER)
0509
0510
0511          FIELD = LIB$EXTZV(0,8,REGISTER)
0512
0513          CALL LINCHK (LUN,2)
```

```
0514
0515            WRITE(LUN,10) REGISTER,ACCS_TYPE(MIN(2,FIELD))
0516    10      FORMAT(' ',T8,'ACCS',T24,Z8.8,/,T40,'ACCELERATOR ',
0517            1 A<COMPRESSC (ACCS_TYPE(MIN(2,FIELD)))>)
0518
0519            IF (FIELD .EQ. 1
0520            1 .OR.
0521            2 FIELD .EQ. 2) THEN
0522
0523            CALL OUTPUT (LUN,REGISTER,V1ACCS,15,15,15,'0')
0524
0525            CALL OUTPUT (LUN,REGISTER,V2ACCS,27,27,29,'0')
0526
0527            CALL OUTPUT (LUN,REGISTER,V3ACCS,31,31,31,'0')
0528            ENDIF
0529
0530            RETURN
0531
0532
0533
0534            ENTRY SBI_FAULTREG (LUN,REGISTER)
0535
0536
0537            CALL LINCHK (LUN,1)
0538
0539            WRITE(LUN,30) REGISTER
0540    30      FORMAT(' ',T8,'SBIFS',T24,Z8.8)
0541
0542            CALL OUTPUT (LUN,REGISTER,V1SBI_FAULT,16,16,19,'0')
0543
0544            CALL OUTPUT (LUN,REGISTER,V2SBI_FAULT,26,26,31,'0')
0545
0546            RETURN
0547
0548
0549
0550            ENTRY SBI_COMPARATOR (LUN,REGISTER)
0551
0552
0553            CALL LINCHK (LUN,1)
0554
0555            WRITE(LUN,40) REGISTER
0556    40      FORMAT(' ',T8,'SBISC',T24,Z8.8)
0557
0558            FIELD = LIB$EXTZV(16,4,REGISTER)
0559
0560            IF (FIELD .NE. 0) THEN
0561
0562            CALL LINCHK (LUN,1)
0563
0564            WRITE(LUN,50) FIELD
0565    50      FORMAT(' ',T40,'COUNT FIELD = ',I2,'.')
0566            ENDIF
0567
0568            FIELD = LIB$EXTZV(20,3,REGISTER)
0569
0570            IF (FIELD .NE. 0) THEN
```

```
0571
0572              CALL LINCHK (LUN,1)
0573
0574              WRITE(LUN,60) SBI_TAG(FIELD)
0575     60       FORMAT(' ',T40,'COMPARE TAG = ',A<COMPRESSC (SBI_TAG(FIELD))>)
0576              ENDIF
0577
0578              FIELD = LIB$EXTZV(23,4,REGISTER)
0579
0580              IF (FIELD .NE. 0) THEN
0581
0582              CALL LINCHK (LUN,1)
0583
0584              WRITE(LUN,70) FIELD
0585     70       FORMAT(' ',T40,'COMPARE COMMAND/MASK = ',I2,'.')
0586              ENDIF
0587
0588              FIELD = LIB$EXTZV(27,2,REGISTER)
0589
0590              IF (FIELD .NE. 0) THEN
0591
0592              CALL LINCHK (LUN,1)
0593
0594              WRITE(LUN,80) COND_LOCK(FIELD)
0595     80       FORMAT(' ',T40,'LOCK = ',A<COMPRESSC (COND_LOCK(FIELD))>)
0596              ENDIF
0597
0598              CALL OUTPUT (LUN,REGISTER,V1SBI_COMPARATR,29,29,31,'0')
0599
0600              RETURN
0601
0602
0603
0604              ENTRY SBI_MAINTENANCE (LUN,REGISTER)
0605
0606
0607              CALL LINCHK (LUN,1)
0608
0609              WRITE(LUN,90) REGISTER
0610     90       FORMAT(' ',T8,'SBIMT',T24,Z8.8)
0611
0612              IF (JIAND(REGISTER,'F05FF900'X) .NE. 0) THEN
0613
0614              CALL LINCHK (LUN,1)
0615
0616              WRITE(LUN,100)
0617     100      FORMAT(' ',T40,'DIAGNOSTIC MODE')
0618              ELSE
0619
0620              IF (JIAND(REGISTER,'200000'X) .EQ. 0) THEN
0621
0622              CALL LINCHK (LUN,1)
0623
0624              WRITE(LUN,105)
0625     105      FORMAT(' ',T40,'SBI INVALIDATE DISABLED')
0626              ENDIF
0627              ENDIF
```

```
0628
0629               RETURN
0630
0631
0632
0633               ENTRY SBI_ERROR (LUN,REGISTER)
0634
0635
0636
0637               CALL LINCHK (LUN,1)
0638
0639               WRITE(LUN,135) REGISTER
0640      135       FORMAT(' ',T8,'SBIER',T24,Z8.8)
0641
0642               CALL OUTPUT (LUN,REGISTER,V1SBI_ERROR,1,1,3,'0')
0643
0644               FIELD = LIB$EXTZV(6,1,REGISTER)
0645
0646               IF (FIELD .NE. 0) THEN
0647
0648               FIELD = LIB$EXTZV(4,2,REGISTER)
0649
0650               CALL LINCHK (LUN,2)
0651
0652               WRITE(LUN,140) IB_STATUS(FIELD)
0653      140       FORMAT(' ',T40,5H IB ',A<COMPRESSC (IB_STATUS(FIELD))>,' TIMEOUT',
0654              1 T40,'IB TIMEOUT')
0655               ENDIF
0656
0657               CALL OUTPUT (LUN,REGISTER,V2SBI_ERROR,7,7,8,'0')
0658
0659               FIELD = LIB$EXTZV(12,1,REGISTER)
0660
0661               IF (FIELD .NE. 0) THEN
0662
0663               Field = LIB$EXTZV(10,2,register)
0664
0665               CALL LINCHK (LUN,2)
0666
0667               WRITE(LUN,150) CP_STATUS(FIELD)
0668      150       FORMAT(' ',T40,A<COMPRESSC (CP_STATUS(FIELD))>,
0669              1 ' TIMEOUT',/,T40,'CPU TIMEOUT')
0670               ENDIF
0671
0672               CALL OUTPUT (LUN,REGISTER,V3SBI_ERROR,13,13,15,'0')
0673
0674               RETURN
0675
0676
0677
0678               ENTRY SBI_TIMEOUT (LUN,REGISTER)
0679
0680
0681
0682               CALL LINCHK (LUN,1)
0683
0684               WRITE(LUN,155) REGISTER
```

```
0685    155    FORMAT(' ',T8,'SBITA',T24,Z8.8)
0686
0687           FIELD = LIB$EXTZV(0,28,REGISTER)
0688
0689           CALL LINCHK (LUN,1)
0690
0691           WRITE(LUN,160) JISHFT(FIELD,2)
0692    160    FORMAT(' ',T40,'TIMEOUT CONSOLE ADDR = ',Z8.8)
0693
0694           CALL OUTPUT (LUN,REGISTER,V1TIMEOUT_ADDR,29,29,29,'0')
0695
0696           FIELD = LIB$EXTZV(30,2,REGISTER)
0697
0698           CALL LINCHK (LUN,1)
0699
0700           WRITE(LUN,170) REF_MODE(FIELD)
0701    170    FORMAT(' ',T40,'TIMEOUT REFERENCE IN ',
0702          1 A<COMPRESSC (REF_MODE(FIELD))>,' MODE')
0703
0704           RETURN
0705
0706
0707
0708           ENTRY SBI_SILO (LUN,REGISTER)
0709
0710
0711
0712           CALL LINCHK (LUN,1)
0713
0714           WRITE(LUN,175) REGISTER
0715    175    FORMAT(' ',T24,Z8.8)
0716
0717           DO 183,J = 0,15
0718
0719           FIELD = LIB$EXTZV(J,1,REGISTER)
0720
0721           IF (FIELD .NE. 0) THEN
0722
0723           CALL LINCHK (LUN,1)
0724
0725           WRITE(LUN,180) J
0726    180    FORMAT(' ',T40,'TR ',I2.2,'. ACTIVE')
0727           ENDIF
0728
0729    183    CONTINUE
0730
0731           FIELD = LIB$EXTZV(16,2,REGISTER)
0732
0733           if (
0734          1 field .ge. 1
0735          1 .and.
0736          1 field .le. 3
0737          1 ) then
0738
0739           CALL LINCHK (LUN,1)
0740
0741           WRITE(LUN,185) SBI_CONFIRM(FIELD)
```

```
0742    185         FORMAT(' ',T40,'CONFIRMATION = '
0743                1 A<COMPRESSC (SBI_CONFIRM(FIELD))>)
0744                    endif
0745
0746                    if (lib$extzv(18,12,register) .ne. 0) then
0747
0748                    FIELD1 = LIB$EXTZV(22,3,REGISTER)
0749
0750                    FIELD = LIB$EXTZV(18,4,REGISTER)
0751
0752                    CALL LINCHK (LUN,1)
0753
0754                    IF (FIELD1 .EQ. COMMAND_ADDRESS) THEN
0755
0756                    WRITE(LUN,187) SBI_FUNCTION(MIN(12,FIELD))
0757    187         FORMAT(' ',T40,'FUNCTION = ',
0758                1 A<COMPRESSC (SBI_FUNCTION(MIN(12,FIELD)))>)
0759
0760                    ELSE IF (FIELD1 .EQ. READ_DATA) THEN
0761
0762                    WRITE(LUN,189) SBI_RESPONSE(MIN(2,FIELD))
0763    189         FORMAT(' ',T40,'DATA READ = ',
0764                1 A<COMPRESSC (SBI_RESPONSE(MIN(2,FIELD)))>)
0765                    ELSE
0766
0767                    WRITE(LUN,191) FIELD
0768    191         FORMAT(' ',T40,'MASK = ',Z1)
0769                    ENDIF
0770
0771                    FIELD = LIB$EXTZV(22,3,REGISTER)
0772
0773                    CALL LINCHK (LUN,1)
0774
0775                    WRITE(LUN,193) SBI_TAG(FIELD)
0776    193         FORMAT(' ',T40,'TAG = ',A<COMPRESSC (SBI_TAG(FIELD))>)
0777
0778                    FIELD = LIB$EXTZV(25,5,REGISTER)
0779
0780                    CALL LINCHK (LUN,1)
0781
0782                    if (field .ne. 16) then
0783
0784                    WRITE(LUN,197) FIELD
0785    197         FORMAT(' ',T40,'ID  = ',I2.2)
0786                    else
0787
0788                    write(lun,198)
0789    198         format(' ',t40,'ID = CPU')
0790                    endif
0791                    endif
0792
0793                    CALL OUTPUT (LUN,REGISTER,V1SBI_SILO,30,30,31,'0')
0794
0795                    RETURN
0796
0797
0798
```

```
0799              ENTRY SBI_COMMAND (LUN,REGISTER)
0800
0801
0802              CALL LINCHK (LUN,1)
0803
0804              WRITE(LUN,200) SBI_FUNCTION(MIN(12,REGISTER))
0805      200     FORMAT(' ',T40,'FUNCTION = ',
0806            1 A<COMPRESSC (SBI_FUNCTION(MIN(12,REGISTER)))>)
0807
0808              RETURN
0809
0810
0811
0812
0813              ENTRY DR780_REGA (LUN,REGISTER)
0814
0815
0816
0817
0818              CALL LINCHK (LUN,2)
0819
0820              WRITE(LUN,209) REGISTER
0821      209     FORMAT(/' ',T8,'"DR" CR',T24,Z8.8)
0822
0823              FIELD = LIB$EXTZV(0,8,REGISTER)
0824
0825              CALL LINCHK (LUN,1)
0826
0827              IF (FIELD .NE. DR780) THEN
0828
0829              WRITE(LUN,210)
0830      210     FORMAT(' ',T40,'ADAPTER NOT "DR"')
0831              ELSE
0832
0833              WRITE(LUN,215)
0834      215     FORMAT(' ',T40,'ADAPTER IS "DR"')
0835
0836              DO 230,J = 2,3
0837
0838              DO 230,I = J*4,(J*4) + 3
0839
0840              FIELD = LIB$EXTZV(I,3,REGISTER)
0841
0842              IF ((FIELD*2)/2 .NE. FIELD) THEN
0843
0844              CALL LINCHK (LUN,2)
0845
0846              WRITE(LUN,220) (J-1),V1DRCR(MIN(3,(FIELD+1)/2)),(J-1)
0847      220     FORMAT(' ',T40,'ID ',I1,'. ERROR'/
0848            1 T40,A<COMPRESSC (V1DRCR(MIN(3,(FIELD+1)/2)))>,' ID ',I1,'.')
0849              ENDIF
0850
0851              if (
0852            1 j .eq. 2
0853            1 .and.
0854            ; i .eq. 8
0855            1 ) then
```

```
0856
0857                call output (lun,register,v2drcr,11,11,11,'0')
0858                endif
0859
0860      230       CONTINUE
0861
0862                CALL OUTPUT (LUN,REGISTER,V3DRCR,15,15,20,'0')
0863
0864                CALL OUTPUT (LUN,REGISTER,V1SBI_REGA,21,21,23,'0')
0865
0866                CALL OUTPUT (LUN,REGISTER,V4DRCR,24,24,24,'0')
0867
0868                CALL OUTPUT (LUN,REGISTER,V2SBI_REGA,26,26,31,'0')
0869                ENDIF
0870
0871                RETURN
0872
0873
0874
0875
0876                ENTRY MS780C_REGA (LUN,REGISTER)
0877
0878
0879
0880                CALL LINCHK (LUN,2)
0881
0882                WRITE(LUN,240) REGISTER
0883      240       FORMAT(/' ',T8,'CSRA',T24,Z8.8)
0884
0885                if (lib$extzv (5,3,register) .ne. 0) then
0886
0887                call linchk (lun,1)
0888
0889                WRITE(LUN,250)
0890      250       FORMAT(' ',T40,'ADAPTER NOT MEMORY TYPE ''C''')
0891                else
0892
0893                CALL OUTPUT (LUN,REGISTER,V1MS780C_REGA,0,0,0,'0')
0894
0895                FIELD = LIB$EXTZV(3,2,REGISTER)
0896
0897                CALL LINCHK (LUN,1)
0898
0899                WRITE(LUN,256) MS780C_RAM_TYPE(FIELD)
0900      256       FORMAT(' ',T40,A<COMPRESS (MS780C_RAM_TYPE(FIELD))>)
0901
0902                IF (FIELD .NE. 0) THEN
0903
0904                IF (FIELD .EQ. FOUR_K) THEN
0905
0906                FIELD = LIB$EXTZV(9,4,REGISTER)
0907
0908                ELSE IF (FIELD .EQ. SIXTEEN_K) THEN
0909
0910                FIELD = LIB$EXTZV(9,6,REGISTER)
0911                ENDIF
0912
```

```
0913              field = (field+1)*64
0914
0915              call linchk (lun,1)
0916
0917              WRITE(LUN,255)
0918       255    FORMAT(' ',T40,'ADAPTER IS MEMORY TYPE ''C''')
0919
0920              CALL LINCHK (LUN,1)
0921
0922              WRITE(LUN,260) FIELD
0923       260    FORMAT(' ',T40,'MEMORY SIZE = ',i<compress4 (field)>,'.K')
0924              ENDIF
0925
0926              CALL OUTPUT (LUN,REGISTER,V1SBI_REGA,21,22,23,'0')
0927
0928              CALL OUTPUT (LUN,REGISTER,V2SBI_REGA,26,26,28,'0')
0929
0930              call output (lun,register,v2sbi_rega,26,30,31,'0')
0931              endif
0932
0933              RETURN
0934
0935
0936
0937
0938              entry ms780e_rega (lun,register)
0939
0940
0941
0942
0943              call linchk (lun,2)
0944
0945              write(lun,261) register
0946       261    format(/' ',t8,'CSRA',t24,z8.8)
0947
0948              if (lib$extzv(5,3,register) .ne. 3) then
0949
0950              call linchk (lun,1)
0951
0952              write(lun,262) 'ADAPTER NOT MEMORY TYPE ''E'''
0953       262    format(' ',t40,a)
0954              else
0955
0956              field = lib$extzv(0,3,register)
0957
0958              if (field .le. 4) then
0959
0960              call linchk (lun,1)
0961
0962              write(lun,263) ms780e_interleave_mode(field)
0963       263    format(' ',t40,a<compressc (ms780e_interleave_mode(field))>)
0964              endif
0965
0966              field = lib$extzv(3,2,register)
0967
0968              call linchk (lun,1)
0969
```

VAX780REG

N 3
16-Sep-1984 00:30:30
5-Sep-1984 14:25:38

VAX-11 FORTRAN V3.4-56
DISK$VMSMASTER:[ERF.SRC]VAX780REG.FOR;1

Page 18

```
0970            write(lun,264) ms780e_ram_type(field)
0971     264    format(' ',t40,a<compressc (ms780e_ram_type(field))>)
0972
0973            call linchk (lun,1)
0974
0975            write(lun,262) 'ADAPTER IS MEMORY TYPE ''E'''
0976
0977            call output (lun,register,v1ms780e_rega,8,8,8,'0')
0978
0979            field = lib$extzv(9,6,register) + 1
0980
0981            call linchk (lun,1)
0982
0983            write(lun,265) field
0984     265    format(' ',t40,'MEMORY SIZE = ',i<compress4 (field)>,'.M')
0985
0986            call output (lun,register,v2ms780e_rega,15,15,20,'0')
0987
0988            call output (lun,register,v1sbi_rega,21,22,23,'0')
0989
0990            call output (lun,register,v2sbi_rega,26,26,28,'0')
0991
0992            call output (lun,register,v2sbi_rega,26,30,31,'0')
0993            endif
0994
0995            return
0996
0997
0998
0999
1000            ENTRY MA780_REGA (LUN,REGISTER)
1001
1002
1003
1004
1005            CALL LINCHK (LUN,2)
1006
1007            WRITE(LUN,269) REGISTER
1008     269    FORMAT(/' ',T8,'PRTCFNG',T24,Z8.8)
1009
1010            FIELD = LIB$EXTZV(0,8,REGISTER)
1011
1012            IF (FIELD .LT. MA780_0 .OR. FIELD .GT. MA780_3) THEN
1013
1014            CALL LINCHK (LUN,1)
1015
1016            WRITE(LUN,275)
1017     275    FORMAT(' ',T40,'ADAPTER NOT MULTI-PORT MEMORY')
1018            ELSE
1019
1020            FIELD = LIB$EXTZV(0,2,REGISTER)
1021
1022            CALL LINCHK (LUN,2)
1023
1024            WRITE(LUN,277) FIELD
1025     277    FORMAT(' ',T40,'ADAPTER IS MULTI-PORT MEMORY',/,
1026           1 T40,'PORT NUMBER = ',I1,'.')
```

```
1027
1028            CALL OUTPUT (LUN,REGISTER,V1SBI_REGA,21,21,23,'0')
1029
1030            CALL OUTPUT (LUN,REGISTER,V2SBI_REGA,26,26,31,'0')
1031            ENDIF
1032
1033            RETURN
1034
1035
1036
1037
1038            entry rh780_configuration_register (lun,register)
1039
1040
1041
1042
1043            call linchk (lun,2)
1044
1045            write(lun,279) register
1046      279   format(/' ',t8,'''RH'' CSR',t24,z8.8)
1047
1048            field = lib$extzv(0,8,register)
1049
1050            call linchk (lun,1)
1051
1052            if (field .ne. mba) then
1053
1054            write(lun,280)
1055      280   format(' ',t40,'ADAPTER NOT MBA')
1056            else
1057
1058            write(lun,285)
1059      285   format(' ',t40,'ADAPTER IS MBA')
1060
1061            call output (lun,register,v1sbi_rega,21,21,23,'0')
1062
1063            call output (lun,register,v2sbi_rega,26,26,31,'0')
1064            endif
1065
1066            return
1067
1068
1069
1070
1071            ENTRY UBA_REGA (LUN,REGISTER)
1072
1073
1074
1075
1076            CALL LINCHK (LUN,2)
1077
1078            WRITE(LUN,289) REGISTER
1079      289   FORMAT(/' ',T8,'''DW'' CSR',T24,Z8.8)
1080
1081            FIELD = LIB$EXTZV(0,8,REGISTER)
1082
1083            CALL LINCHK (LUN,1)
```

```
1084
1085              IF (FIELD .LT. UBA_0 .OR. FIELD .GT. UBA_3) THEN
1086
1087              WRITE(LUN,295)
1088      295     FORMAT(' ',T40,'ADAPTER NOT UBA')
1089              ELSE
1090
1091              FIELD = LIB$EXTZV(0,2,REGISTER)
1092
1093              WRITE(LUN,300) FIELD
1094      300     FORMAT(/' ',T40,'ADAPTER IS UBA ',I1,'.')
1095
1096              CALL OUTPUT (LUN,REGISTER,V1UBA_REGA,16,16,18,'0')
1097
1098              CALL OUTPUT (LUN,REGISTER,V1SBI_REGA,21,21,23,'0')
1099
1100              CALL OUTPUT (LUN,REGISTER,V2SBI_REGA,26,26,31,'0')
1101              ENDIF
1102
1103              RETURN
1104
1105
1106
1107
1108              entry ci780_rega (lun,register)
1109
1110
1111
1112
1113              call linchk (lun,2)
1114
1115              write(lun,400) register
1116      400     format(/' ',t8,'CNFGR',t24,z8.8)
1117
1118              call linchk (lun,1)
1119
1120              if (lib$extzv(0,8,register) .ne. '38'x) then
1121
1122              write(lun,405) 'ADAPTER NOT ''CI'''
1123      405     format(' ',t40,a)
1124              else
1125
1126              write(lun,405) 'ADAPTER IS ''CI'''
1127
1128              call output (lun,register,v1ci780_rega,8,8,10,'0')
1129
1130              call output (lun,register,v2ci780_rega,16,16,20,'0')
1131
1132              call output (lun,register,v1sbi_rega,21,22,23,'0')
1133
1134              call output (lun,register,v2sbi_rega,26,26,27,'0')
1135
1136              call output (lun,register,v2sbi_rega,26,29,31,'0')
1137              endif
1138
1139              return
1140
```

1141   END

PROGRAM SECTIONS

    Name   Bytes   Attributes

  0 $CODE   5228   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA   1474   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL   5168   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG

    Total Space Allocated   11870

ENTRY POINTS

    Address Type Name   Address Type Name

  0-00000010   ACCS_780   0-000010F7   CI780_REGA
  0-000008BA   DR780_REGA   0-00000E45   MA780_REGA
  0-00000A73   MS780C_REGA   0-00000C2F   MS780E_REGA
  0-00000F3E   RH780_CONFIGURATION_REGISTER 0-0000085C   SBI_COMMAND
  0-0000013B   SBI_COMPARATOR   0-0000036A   SBI_ERROR
  0-000000D1   SBI_FAULTREG   0-000002C6   SBI_MAINTENANCE
  0-000005B0   SBI_SILO   0-000004BB   SBI_TIMEOUT
  0-00001000   UBA_REGA   0-00000000   VAX780REG

VARIABLES

    Address Type Name   Address Type Name   Address Type Name   Address Type Name

  2-00000A4C I*4 FIELD   2-00000A50 I*4 FIELD1   2-00000A58 I*4 I   2-00000A54 I*4 J
  AP-00000004@ L*1 LUN   2-00000A5C@ I*4 REGISTER

ARRAYS

    Address Type Name   Bytes Dimensions

  2-0000056D CHAR ACCS_TYPE   45   (0:2)
  2-00000820 CHAR COND_LOCK   75   (3)
  2-00000000 CHAR CP_STATUS   88   (0:3)
  2-00000000 CHAR IB_STATUS   88   (0:3)
  2-00000216 CHAR MS780C_RAM_TYPE   104   (0:3)
  2-000002E6 CHAR MS780E_INTERLEAVE_MODE   155   (0:4)
  2-0000027E CHAR MS780E_RAM_TYPE   104   (0:3)
  2-00000703 CHAR REF_MODE   44   (0:3)
  2-00000471 CHAR SBI_CONFIRM   36   (3)
  2-0000086B CHAR SBI_FUNCTION   299   (0:12)
  2-0000072F CHAR SBI_RESPONSE   63   (0:2)
  2-00000790 CHAR SBI_TAG   144   (0:7)
  2-00000000 CHAR TIMEOUT_STATUS   88   (0:3)
  2-0000059A CHAR V1ACCS   20   (15:15)
  2-00000996 CHAR V1CI780_REGA   57   (8:10)

```
2-00000426   CHAR V1DRCR                          75   (3)
2-00000112   CHAR V1MS780C_REGA                   17   (0:0)
2-00000112   CHAR V1MS780E_REGA                   17   (8:8)
2-000006B2   CHAR V1SBI_COMPARATR                 81   (29:31)
2-00000495   CHAR V1SBI_ERROR                     66   (3)
2-000005F9   CHAR V1SBI_FAULT                     92   (16:19)
2-00000655   CHAR V1SBI_REGA                      93   (21:23)
2-0000076E   CHAR V1SBI_SILO                      34   (30:31)
2-00000550   CHAR V1TIMEOUT_ADDR                  29   (29:29)
2-00000123   CHAR V1UBA_REGA                      63   (16:18)
2-000005AE   CHAR V2ACCS                          69   (27:29)
2-000009CF   CHAR V2CI780_REGA                   125   (16:20)
2-00000381   CHAR V2DRCR                          24   (11:11)
2-00000162   CHAR V2MS780E_REGA                  180   (15:20)
2-000004D7   CHAR V2SBI_ERROR                     46   (7:8)
2-00000058   CHAR V2SBI_FAULT                    186   (26:31)
2-00000058   CHAR V2SBI_REGA                     186   (26:31)
2-000005F3   CHAR V3ACCS                           6   (31:31)
2-00000399   CHAR V3DRCR                         126   (15:20)
2-00000505   CHAR V3SBI_ERROR                     75   (13:15)
2-00000417   CHAR V4DRCR                          15   (24:24)
```

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-000000CE	10'	1-000000F6	30'	1-00000108	40'	1-0000011A	50'	1-00000135	60'	1-00000151	70'
1-00000175	80'	1-0000018A	90'	1-0000019C	100'	1-000001B3	105'	1-000001D2	135'	1-000001E4	140'
1-0000020F	150'	1-00000235	155'	1-00000247	160'	1-00000269	170'	1-00000293	175'	1-0000029C	180'
**	183	1-000002B4	185'	1-000002D1	187'	1-000002EA	189'	1-00000304	191'	1-00000315	193'
1-00000329	197'	1-0000033A	198'	1-0000034A	200'	1-00000363	209'	1-00000378	210'	1-00000390	215'
1-000003A7	220'	**	230	1-000003D1	240'	1-000003E3	250'	1-00000412	255'	1-00000406	256'
1-00000434	260'	1-00000454	261'	1-00000466	262'	1-0000046D	263'	1-00000479	264'	1-00000485	265'
1-000004A5	269'	1-000004BA	275'	1-000004DF	277'	1-0000051B	279'	1-00000531	280'	1-00000548	285'
1-0000055E	289'	1-00000574	295'	1-0000058B	300'	1-000005A8	400'	1-000005BB	405'		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	COMPRESSC	I*4	LIB$EXTZV		LINCHK		OUTPUT

COMMAND QUALIFIERS

```
FORTRAN /LIS=LIS$:VAX780REG/OBJ=OBJ$:VAX780REG MSRC$:VAX780REG

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19
```

COMPILATION STATISTICS

    Run Time: 13.57 seconds
    Elapsed Time: 29.06 seconds
    Page Faults: 278
    Dynamic Memory: 290 pages

EVLDEF
MDL

LIBTAIL
B32

XXXQUE
LIS

UTILDEF
REQ

ERRDEF
MDL

EVCDEF
MDL

CONSOLE
LIS

USEROPEN
LIS

VAXPSL
LIS

ERRFMT

VAX780REG
LIS

ERRFMT
MAP

EVLIBRARY
B32

VAX750REG
LIS

VECMAPREG
LIS

ERRFMT
LIS

EVL

LIBHEAD
B32

EVL
MAP

WQDEF
MDL